Quantum compilation and its noise resilience

Sumeet Khatri

Department of Physics & Astronomy, Louisiana State University

Slides for [Quantum 3, 140 (2019)] and [NJP 22, 043006 (2020)]

December 18, 2020
Very soon, quantum computers will be developed. The ultimate goal is to create fault-tolerant quantum computers. But this requires many good-quality qubits: \( \approx 1000 \) physical qubits per logical qubit.

We currently have noisy intermediate-scale quantum (NISQ) computers:

- Limited number of qubits (\( \approx 50-100 \)), not fault tolerant.
- Limited connectivity between qubits.
- Limited circuit depth due to noise.

What can we do with such quantum computers?
Variational quantum algorithms [NJP 18, 023023 (2016)]

▶ Variational quantum eigensolver (VQE).
[Nat. Commun. 5, 4213 (2014)]

▶ Variational quantum autoencoders.
[Quant. Sci. Technol. 2, 045001 (2017)]

▶ Variational quantum state diagonalization.
[npj Quant. Inf. 5, 57 (2019)]

★ Our variational algorithm: Quantum-assisted quantum compiling
[Quantum 3, 140 (2019)].
Compiling quantum algorithms

To run an algorithm on a quantum computer, we must *compile it*. 
Compiling quantum algorithms

To run an algorithm on a quantum computer, we must *compile it*.

Need to adapt algorithms to NISQ device constraints:

- Native gate alphabet;
- Connectivity;
- Short depth (use fewest possible gates).

Note: short depth is also important for future fault-tolerant quantum computers.
Compiling quantum algorithms

Typical compilation procedure

[IEEE Trans. CAD 27(3), 436 (2008)]
[Quantum Sci. Technol. 3, 025004 (2018)]

**Given:** Unitary $U$ as a high level gate sequence.

**Step 1:** Individually decompose every gate into the native gate alphabet.

**Step 2:** Sweep through the circuit, removing and/or combining redundant gates via, e.g., matrix identities.

**Output:** Gate sequence $V$ (compilation of $U$).
Compiling quantum algorithms

Typical compilation procedure

[IEEE Trans. CAD 27(3), 436 (2008)]
[Quantum Sci. Technol. 3, 025004 (2018)]

**Given:** Unitary $U$ as a high level gate sequence.

**Step 1:** Individually decompose every gate into the native gate alphabet.

**Step 2:** Sweep through the circuit, removing and/or combining redundant gates via, e.g., matrix identities.

**Output:** Gate sequence $V$ (compilation of $U$).

**Drawbacks**

- Step 2 can be inefficient, esp. for large circuits.
- Requires classical simulation of the circuit in general.
Compiling quantum algorithms

**Typical compilation procedure**

[IEEE Trans. CAD 27(3), 436 (2008)]

[Quantum Sci. Technol. 3, 025004 (2018)]

**Given:** Unitary $U$ as a high level gate sequence.

**Step 1:** Individually decompose every gate into the native gate alphabet.

**Step 2:** Sweep through the circuit, removing and/or combining redundant gates via, e.g., matrix identities.

**Output:** Gate sequence $V$ (compilation of $U$).

**Drawbacks**

- Step 2 can be inefficient, esp. for large circuits.
- Requires classical simulation of the circuit in general.

- **Quantum-assisted quantum compiling (QAQC)** is a method for quantum compiling that runs the algorithms on an actual quantum computer, and can perform *approximate optimal compilation*. 

![Diagram](n-qubit quantum Fourier transform)
Compiling quantum algorithms

Typical compilation procedure

[IEEE Trans. CAD 27(3), 436 (2008)]
[Quantum Sci. Technol. 3, 025004 (2018)]

**Given:** Unitary $U$ as a high level gate sequence.

**Step 1:** Individually decompose every gate into the native gate alphabet.

**Step 2:** Sweep through the circuit, removing and/or combining redundant gates via, e.g., matrix identities.

**Output:** Gate sequence $V$ (compilation of $U$).

**Drawbacks**

- Step 2 can be inefficient, esp. for large circuits.
- Requires classical simulation of the circuit in general.

- **Quantum-assisted quantum compiling (QAQC)** is a method for quantum compiling that runs the algorithms on an actual quantum computer, and can perform *approximate optimal compilation*. 

\[ n \text{-qubit quantum Fourier transform} \]

\[ \begin{array}{c}
H & R_2 & \cdots & R_n \\
\vdots & \cdots & \cdots & \cdots \\
& \vdots & \cdots & \cdots \\
& & \vdots & \cdots \\
& & & H \\
\end{array} \]
Quantum-assisted quantum compiling (QAQC) [Quantum 3, 140 (2019)]

QAQC performs optimal (approximate) algorithm depth compression

- \( U \) is the result of Step 1: an initial “pre-compiled”, non-optimal gate sequence.

- \( V \) is the result of QAQC.

Algorithm depth compression
Quantum-assisted quantum compiling (QAQC) [Quantum 3, 140 (2019)]

**Given:** A unitary $U$.

**Goal:** find the optimal parameterized gate sequence $V_k(\alpha)$.

$V_k(\alpha) = \ldots \approx U$

- $C(U, V_k(\alpha))$ quantifies the distance between $U$ and $V_k(\alpha)$.
- $\alpha \rightarrow \alpha'$ via gradient descent optimizer (or non-gradient optimizer); $k \rightarrow k'$ via simulated annealing (or genetic algorithms).
Quantum-assisted quantum compiling (QAQC) [Quantum 3, 140 (2019)]

Cost function calculation on quantum computer

\[
|0\rangle \ H \ U \ H \ |0\rangle \\
|0\rangle \ H \ |0\rangle \\
\vdots \\
|0\rangle \ H \ |0\rangle \\
|0\rangle \ V^* \ |0\rangle \\
\vdots \\
|0\rangle \\
\text{Hilbert-Schmidt Test (HST)}
\]

- Probability of the all-zeros outcome is
  \[
  \frac{1}{2^{2n}} |\text{Tr}[V^* U]|^2.
  \]
  The quantity is invariant under global phase.

* Can also compute gradient (see paper for details).
Cost function calculation on quantum computer

![Quantum circuit diagram]

- Probability of the all-zeros outcome is
  \[
  \frac{1}{2^{2n}} |\text{Tr}[V^\dagger U]|^2.
  \]

  The quantity is invariant under global phase.

- Cost function:
  \[
  C_{\text{HST}}(U, V) := 1 - \frac{1}{2^{2n}} |\text{Tr}[V^\dagger U]|^2.
  \]

★ Can also compute gradient (see paper for details).
Cost function calculation on quantum computer

- Probability of the all-zeros outcome is
  \[ \frac{1}{2^n} |\text{Tr}[V^\dagger U]|^2. \]
  The quantity is invariant under global phase.

- Cost function:
  \[ C_{\text{HST}}(U, V) := 1 - \frac{1}{2^n} |\text{Tr}[V^\dagger U]|^2. \]

- Related to trace distance:
  \[ \frac{1}{2} \left\| |\Phi_U\rangle\langle \Phi_U| - |\Phi_V\rangle\langle \Phi_V| \right\|_1 = \sqrt{C_{\text{HST}}(U, V)}, \]
  \[ |\Phi_U\rangle = (\mathbb{I} \otimes U)|\Phi\rangle, \]
  \[ |\Phi_V\rangle = (\mathbb{I} \otimes V)|\Phi\rangle, \]
  \[ |\Phi\rangle: n\text{-qubit maximally entangled state}. \]

- Can also compute gradient (see paper for details).
Alternate (local) cost function for large problem sizes

Local Hilbert-Schmidt Test (LHST)

★ Can also compute gradient (see paper for details).

Cost function:

\[ C_{\text{LHST}}(U, V) := \frac{1}{n} \sum_{j=1}^{n} (1 - \text{Pr}[0, 0]) \]

★ Faithful cost function (like HST).

★ Works better for optimization on large problem sizes.
Advantages of QAQC

- No need to classically simulate the circuit.
- Both cost function calculations difficult to simulate classically.
  - Both are DQC1-hard [PRL 81, 5672 (1998)].
  - Classical simulation of DQC1 is impossible unless polynomial hierarchy collapses to the second level [PRL 120, 200502 (2018)].
- Holistic approach: optimizes gate parameters and also structure, and does not compile in gate-by-gate manner.
- Not just a NISQ application: depth compression is useful even for fault-tolerant quantum computers!

How well does it perform, especially in the presence of noise?
Quantum-assisted quantum compiling (QAQC) [Quantum 3, 140 (2019)]

Results on a noisy simulator

\[
\begin{align*}
R_z(\alpha_1) & \quad R_z(\alpha'_1) \\
R_z(\alpha_2) & \quad R_z(\alpha'_2) \\
R_z(\alpha_3) & \quad R_z(\alpha'_3) \\
R_z(\alpha_4) & \quad R_z(\alpha'_4)
\end{align*}
\]

Observations

★ HST does not optimize well, but LHST does.
★ LHST optimization finds the correct minimum even in the presence of noise!
Resilience is with respect to the gate parameters: optimal gate parameters obtained via noisy optimization are also optimal without noise. So we obtain the true optimum even with noise.

\[ \text{opt (noisy)} \]
\[ \text{opt (w/o noise)} \]

We provide proofs of noise resilience under certain noise models.
Alternate HST circuit
We prove noise resilience under this noise model. We can also add noise in between $U$ and $V^\dagger$ (see paper for details).

Analogous results for LHST circuit.
We prove noise resilience under this noise model. We can also add noise in between $U$ and $V^\dagger$ (see paper for details).

Analogous results for LHST circuit.
Noise resilience [NJP 22, 043006 (2020)]

Examples
Variational algorithm (QAQC) to compile quantum algorithms ⇒ depth reduction. 
[Quantum 3, 140 (2019)]
- Circuits (HST and LHST) that compare two quantum circuits.

Both the HST and LHST cost functions exhibit optimal parameter resilience under certain noise models. 
[NJP 22, 043006 (2020)]
- Optimal solution under noisy optimization corresponds to an optimal solution under noiseless optimization.
Summary & outlook

▶ Variational algorithm (QAQC) to compile quantum algorithms ⇒ depth reduction.

[Quantum 3, 140 (2019)]

▶ Circuits (HST and LHST) that compare two quantum circuits.

▶ Both the HST and LHST cost functions exhibit optimal parameter resilience under certain noise models.

[NJP 22, 043006 (2020)]

▶ Optimal solution under noisy optimization corresponds to an optimal solution under noiseless optimization.

Future directions

▶ Investigate noise resilience of other variational algorithms.

▶ Algorithms for error mitigation on NISQ devices.